1 PRNG

1.1 Linear Congruential Generator

The Linear Congruential Generator is given by Eq. 1

$$X_{n+1} = (aX_n + c) mod m \tag{1}$$

where X is the sequence of pseudo-random values m, $0 < m$ gives the PRNG space, a, $0 < a < m$ is multiplier, c, $0 \leq c < m$ is increment and x_0, $0 \leq x_0 < m$ the seed or start value.

1.2 Blum Blum Shub

Blum Blum Shub is given by Eq. 2

$$x_{n+1} = x_n^2 \bmod M \tag{2}$$

where $M = pq$ is the product of two large primes p and q. At each step of the algorithm, some output is derived from x_{n+1}; the output is commonly either the bit parity of x_{n+1} or one or more of the least significant bits of x_{n+1}. The seed x_0 should be an integer that is co-prime to M (i.e. p and q are not factors of x_0) and not 1 or 0.

The two primes, p and q, should both be congruent to $3 \pmod 4$ (this guarantees that each quadratic residue has one square root which is also a quadratic residue), and should be safe primes with a small $gcd((p-3)/2, (q-3)/2)$ (this makes the cycle length large).

1.3 Linear-feedback shift register

A linear-feedback shift register (LFSR) is a register of bits that performs discrete step operations that: shifts the bits one position to the left and replaces the vacated bit by the exclusive

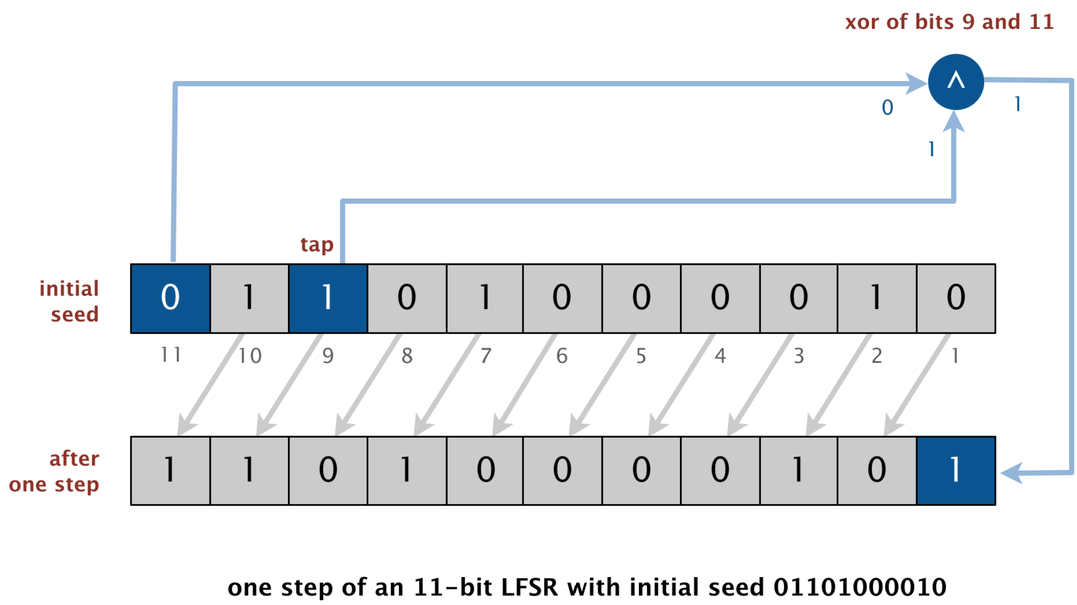

Figure 1: LFSR

or(xor) of the bit shifted off and the bit previously at a given tap position in the register. A

LFSR has three parameters that characterize the sequence of bits it produces: the number of bits n, the initial seed (the sequence of bits that initializes the register), and the tap position tap. As in the example in Lecture 0, the Fig. 1 illustrates one step of an 11-bit LFSR with initial seed 01101000010 and tap positions 9.

1. Perform the following task for all the above PRNG
   (a) Write python code (provide correct input and output)
   (b) You can test the period of an PRNG for a given seed by count the number of iterations of the PRNG need to generate the seed value once more. Write a python code to find the periods of all the above PRNGs.
2. What are the requirements for a cryptographically secure PRNG?